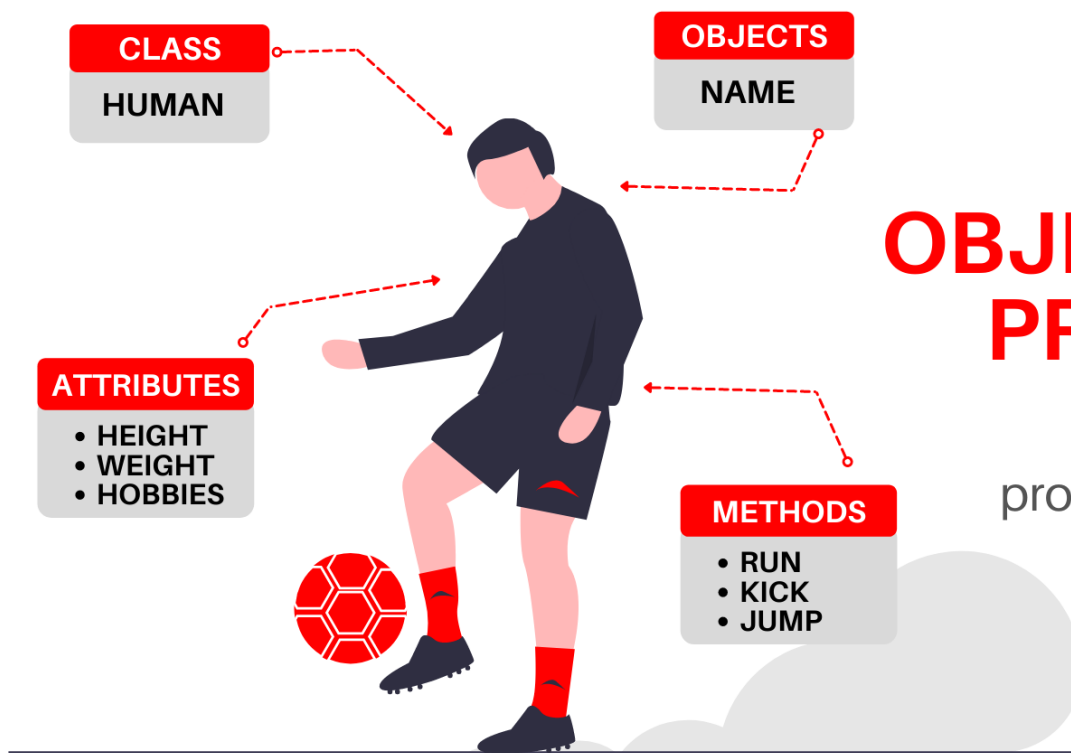


Introduction

Object-oriented programming

- Object-oriented programming: As the name suggests, Object-Oriented Programming or OOPs refers to languages that uses objects in programming. Object-oriented programming aims to implement real-world entities like inheritance, hiding, polymorphism etc in programming.
- The main aim of OOP is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function.

Object-oriented programming



OBJECT-ORIENTED PRORAMMING

What is object-oriented programming and what are the advantages of it?

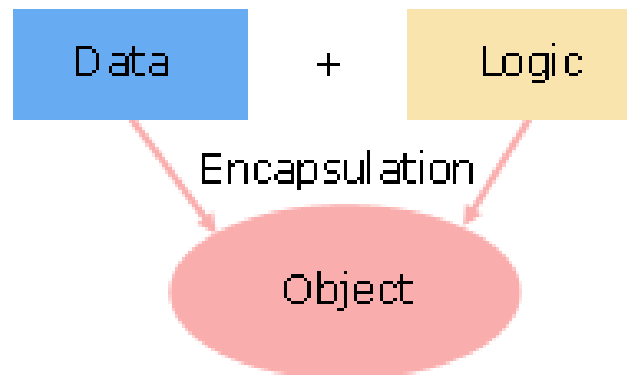
Object Terminology

There are **four fundamental** concepts

- Encapsulation
- Hierarchy
- Abstraction
- Polymorphism

Encapsulation

- Encapsulation **separates** the **implementation details** of an object from its **external appearance**.
- Encapsulation **focuses** on the **interior** of an object, combining the data that describes the object's state and the algorithms that **define** its **behavior**.



Encapsulation

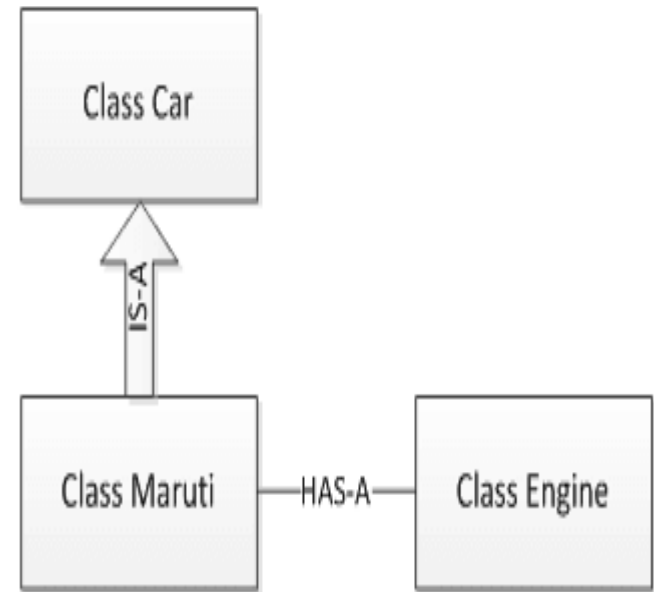
- A **well-encapsulated** object has all of its implementation details hidden within the object.
- If an object is well-encapsulated, a developer can **change** the object's **internal structure** without introducing any **changes** to the **software** that uses the object.

| class Student |
|--|
| char enroll[10]; char name[38]; double gpa; |
| void setEnroll(char cER[]); public: void setName(char cName[]); void setGpa(double dGpa); char[] getEnroll(); char[] getName(); double getGpac(); ... |

Hierarchy (Inheritance)

Some of the objects in an application may be hierarchically related to one another. The hierarchy may be one of:

- aggregation, or
- shared structure and behavior
- Aggregation describes a "has a" relationship between objects. The parent object "has a" child object. The two objects need not share a common structure.
- Shared structure and behavior entails an "is a kind of" relationship. This appears as a hierarchy of classes. One class "is a kind of" another class



Abstraction

- Data Abstraction is the property by virtue of which only the essential details are displayed to the user. The trivial or the non-essentials units are not displayed to the user. Ex: A car is viewed as a car rather than its individual components.
- Data Abstraction may also be defined as the process of identifying only the required characteristics of an object ignoring the irrelevant details. The properties and behaviors of an object differentiate it from other objects of similar type and also help in classifying/grouping the objects.

Abstraction



Polymorphism

Polymorphism relates the implementation for an object based on its type



Summary

- Objects are abstractions of the most important chunks of information from a problem domain. They distinguish the different feature sets in the problem domain.
- A class describes the structure common to a set of similar objects. Each object in the set is a single instance of its class.
- Encapsulation hides the implementation details within a class - the internal data and internal logic are invisible to client applications that use objects of that class.
- We can upgrade the structure of a well-encapsulated class without altering any client code.
- The cornerstones of object-oriented programming are encapsulation, inheritance and polymorphism.